Lab 5. Arrays – part 1

Lab solutions

Starting this week, we will make the lab solutions available as a zip file that you can directly import in Eclipse. The instructions how to do this will be available on eLearning.

Homework

The homework for this week consists of 8 multiple choice questions organised as a test in eLearning. You need to answer them online, via eLearning. Login to eLearning, go to ***Course Material*** (left panel) and click on the link ***w5-homework*** to start the online test (homework). And you already know the rules and hints:

1. Do not delay the submission of the homework till the very last moment as eLearning may get overloaded and you may miss the deadline! You will not be able to submit after the deadline.

2. If you have problems submitting in eLearning before the deadline, email your answers to sit.info1103@sydney.edu.au (e.g. 1A, 2B, etc.).

3. You are allowed 2 attempts. If you submit twice, the second attempt will be marked.

4. Incomplete submissions are not valid submissions and are ignored. An incomplete submission is a submission started before the deadline but not submitted before the deadline. Only complete (valid) submissions are marked. Start working on the homework early so that you can finish it before the deadline, and remember to press "Submit"!

Here are the questions for your information (remember that they must be answered online via eLearning):

1. Which is the correct statement for creating an array `myArray` of ten integer elements?

A) `int[] myArray = new int[9];`
B) `int[] myArray = new int[10];`
C) `int[10] myArray;`
D) `int myArray[10];`

2. What will be printed?

```
int[] myArray = { 10, 20, 30, 40, 50 };
System.out.print(myArray[2]);
System.out.print(myArray[3]);
```

A) 1050
B) 2030
C) 3040
D) 4050

3. Which statement is correct?

```
int[] someArray = new int[6];
for (int i = 1; i < 6; i++)
{
   someArray[i] = i + 1;
}
```

A) The `for` loop initializes all the elements of the array.
B) The `for` loop initializes all the elements except the first element.
C) The `for` loop initializes all the elements except the last element.
D) The `for` loop initializes all the elements except the first and last elements.

4. What is the result of executing this code?

```
int[] marks = { 90, 45, 67 };
for (int i = 0; i <= 3; i++)
{
   System.out.print(marks[i]+ " ");
}
```

A) The code will not give any output.
B) The code will print: 90 45 67 .
C) The code will cause an "out of bounds" error.
D) The code will execute an infinite loop.

5. What will happen if we use `size()` to check the length of the array instead of using the correct method `length()`?

A) The program will crash.
B) The program will not compile.
C) The program will run but will produce an uncertain result.
D) The compiler will automatically correct the error.

6. Which one of the following is the correct header for a method named `arrMethod` that is called like this:

```
arrMethod(intArray);  // intArray is an integer array of size 3
```

A) `public static void arrMethod(int[] ar, int n)`
B) `public static void arrMethod(r[], n)`
C) `public static void arrMethod(int n , int[] ar)`
D) `public static void arrMethod(int[] ar)`

7. What will be printed?

```
int[][] arr =
{
   { 3, 2, 3 },
   { 0, 0, 0 }
};
System.out.print(arr[0][0]);
System.out.print(arr[1][0]);
```

A) 00
B) 31
C) 30
D) 03

8. What will be the value of `numArray[1][2]`after this code is executed:

```
int count = 0;
int[][] numArray = new int[2][3];
for (int i = 0; i < 3; i++)
{
   for (int j = 0; j < 2; j++)
   {
      numArray[j][i] = count;
      count++;
   }
}
```

A) 2
B) 5
C) 3
D) 4

Mid-semester survey

The survey is administered by the School of IT. You will need to complete it for all School of IT courses.

How is this course going so far? Any problems? Any suggestions for improvement? The focus of the survey is to identify issues and make changes to address them before the end of the semester. Please give us some constructive feedback, we take your comments seriously. The survey is anonymous.

Please log in to eLearning and complete the mid-semester survey. It is located in the menu on the left. This should take about 10 minutes. And thank you for completing the survey!

1. Initializing an array

Write a program in a class `ArrayInitialization.java` that creates an array of 10 integer values and then initializes these values to:

a) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
b) 0, 2, 4, 6, 8, 10, 12, 14, 16, 18
c) 1, 4, 9, 16, 25, 36, 49, 64, 81, 100

Write your code in the `main` method and use a `for` loop for the initialization (one loop for each case).

2. Printing an array in reverse order

Write a program in a class `ReverseArray.java` that reads 10 integer numbers from the keyboard, one per line, stores them in an array and then prints the numbers in the opposite order from the order they were entered. Use one loop for entering the numbers and another one for printing them. Write your code in the `main` method. Here is an example output:

```
Enter element 1:
6
Enter element 2:
19
Enter element 3:
5
Enter element 4:
7
Enter element 5:
9
The entered numbers in reverse order are: 9 7 5 19 6
```

3. Temperature above average

Write a program in a class `NumberAboveAverage.java` that counts the number of days when the temperature was above average. Create an array to store 10 temperatures and fill it in with values using {}. Compute the average temperature and then count and print the number of days when the temperature was above this average.

If the array contains the values {18.3, 20.4, 19.6, 22, 24, 24.6, 21.3, 19.2, 17.1, 18.6}, a sample output of the program is:

```
The average temperature is: 20.509999999999998
Day 3 had temperature 22.0 which was above average
Day 4 had temperature 24.0 which was above average
Day 5 had temperature 24.6 which was above average
Day 6 had temperature 21.3 which was above average
The number of days with a temperature above average is: 4
```

4. Check Array

Write a program in a class `CheckArray.java` that checks if an array of n integer elements contains the numbers from 0 to n-1.

1) Write a method `check` with the following header and javadoc comments:

```
/**
    * Checks if a given array arr with size n contains the numbers from 0 to n-1
    * @param arr - array with n integer elements
    * @return true if arr contains the numbers from 0 to n-1 (all numbers, and just once)
    * and false otherwise
    */
   public static boolean check(int[] arr)
```

2) In the `main` method, create several arrays that satisfy and do not satisfy the condition. Then call the method `check` several times, each time passing one array as a parameter. Check if the array satisfies the condition and print an appropriate message. For example:

   If the array contains the values `{ 2, 1, 0, 4, 5, 3 }`, your program should print: `The array contains the numbers from 0 to n-1`

If the array contains the values `{ 2, 8, 0, 4, 5, 3 }`, your program should print: `The array does not contain the numbers from 0 to n-1`

If the array contains the values `{ 1, 2, 3, 4, 5, 6 }`, your program should print: `The array does not contain the numbers from 0 to n-1`

5. Alternating sum

Write a program in a class `AlternatingSum.java` that computes the alternating sum of all elements in the array. For example, if the array is {2, 6, 17, 8}, the alternating sum is 2-6+17-8.

In the `main` method create an array of type `double` and fill it in with values using {}. Your program should include two other methods: `alternatingSum` and `printArray` that are called from the `main` method. Their headers and `javadoc` comments are:

```
/**
      Computes the alternating sum of the values in an array, e.g.
      if the array is {1 6 7 9}, the sum will be 1-6+7-9
      @param data - an array
      @return the alternating sum of the values in data
   */
   public static double alternatingSum(double[] data)
```

```
/**
   Prints the array elements using | as separator
   @param data - an array
*/
   public static void printArray(double[] data)
```

An example output of your program is:

```
Array:
1.6 | 4.2 | 9.3 | 16.0 | 9.0 | 7.4 | 4.0 | 9.0 | 11.2
Alternating sum: -1.5
```

6. Filling and printing of a 2D array

Write a program in a class `FillArray.java.` It should read values of type `double` from the keyboard and place them in a 2D array and then print this array nicely. You can think of the array as a picture, where each array element corresponds to a pixel and the value of each array element corresponds to a colour value.

- Declare two constants: N=2 for the number of rows and M=3 for the number of columns; you can change these values if you want.
- Write the following two methods:

```
/**
   Reads double values from the keyboard and fills in an N x M array
   @return the N x M array
*/
public static double[][] readPicture()
```

```
/**
   Prints a 2D array of double values using %6.2f
   @param ar - the array to print
*/
public static void printArray(double[][] ar)
```

- Call these two methods from the `main` method.

Some sample output is here:

```
Please enter 6 numbers:
1 23 4 56 6 8

The picture is:
  1.00  23.00   4.00
 56.00   6.00   8.00


Please enter 6 numbers:
12 34
89 765
3
9

The picture is:
 12.00  34.00  89.00
765.00   3.00   9.00
```

As you can see the `readInput` method should read the user input no matter if the user enters the numbers on one line or more than one lines.

Note: In the challenge question we will extend this program!

7. Sum of elements above the main diagonal

Write a program in a class `AboveMainDiag.java` that finds the sum of all elements above the main diagonal in an NxN array.

Declare a constant N=4 for the size of the array (i.e. the number of rows and columns).

- In the `main` method, use two nested `for` loops to initialize the array to integer random numbers between 1 and 100. Hint: there is an example in the lecture slides. In addition to initialization, also print the elements in the array from the same nested loops.
- Write a method called `sumAboveMainDiag` that takes the array as an argument and returns the sum of its elements above the main diagonal. Hint: What condition is true for the indices of the elements above the main diagonal? Write them down and find the relation between the two indices!
- Call this method from the `main` method and print the sum that it returns.

An example output is here:

```
75 64 58 90
92 52 23 96
64 70 39 1
92 8 20 86
The sum of the elements above the main diagonal is: 332
```

8. Challenge question

This question is an extension of question 6. Write a program in a class `ArrayFilter.java.` It should:

1) read values of type `double` from the keyboard to fill in the 2D array `picture`
2) print this array and
3) compute a filtered version of the array, called `newPicture` where each element is either 0 or 1 depending on a threshold. This threshold is computed by computing the average of all values of `picture` and multiplying it by a given threshold. If the value of an element of `picture` is above the threshold, the corresponding element in `newPicture` should be 1, otherwise it should be 0.

In question 6 we already wrote the methods for tasks 1) and 2); use them.

Some example outputs are here:

```
Please enter 6 numbers:
10.3 23.4 56.7 89.1 76.4 34.2

Old picture:
10.30 23.40 56.70
89.10 76.40 34.20

The average is 48.35; anything above 38.68 will be 1; anything below it will be 0.

New picture:
0.00 0.00 1.00
1.00 1.00 0.00
```

The following skeleton code is given (download it from eLearning); your task is to write the method `applyThreshold.`

```
import java.util.Scanner;

public class ArrayFilter {
       public static final int N = 2; //number of rows
       public static final int M = 3; //number of columns
       public static final double FACTOR = 0.8; //factor for multiplication

       public static void main(String[] args) {
             double[][] picture = readPicture();
             System.out.println("\nOld picture:");
             printArray(picture);
             double[][] newPicture = applyThreshold(picture, FACTOR);
             System.out.println("\nNew picture:");
             printArray(newPicture);
       }

       /**
          Reads double values from the keyboard and fills in a N x M array
          @return the N x M array
       */
       public static double[][] readPicture() {
              // written in Question 6
       }


       /**
          Takes a 2D array (pic) and a coefficient. Returns newPic - a filtered version
```

```
         of pic, whose elements are either 0 or 1 depending on a threshold.
         Threshold = coefficient * average value of all elements in pic.
         The value of newPic will be 1, if the corresponding value in pic is above the
         threshold and 0 otherwise.
         Prints the message “The average is … etc”
         @param pic - the original array (picture)
         @param coefficient - the coefficient used to compute the threshold
         @return the new filtered picture
      */
      public static double[][] applyThreshold(double[][] pic, double coefficient) {

       // write this method
      }

      /**
         Prints a 2D array of double values using %.2f
         @param ar - the array to print
      */
      public static void printArray(double[][] ar) {

         // written in Question 6
      }

}
```